

# От алгоритма к программе

Чтобы выполнить алгоритм на ЭВМ, надо представить его в виде текста на языке программирования, т.е. в виде программы.

Программа включает в себя кроме алгоритма описание всех данных, используемых алгоритмом.

Программа, введенная в ЭВМ, обрабатывается транслятором соответствующего языка.

В процессе трансляции проверяется правильность записи конструкций языка (синтаксический контроль).

Если обнаружены ошибки, выводится соответствующее сообщение и трансляция приостанавливается, если ошибок нет - строится выполняемая программа.

При этом выделяется память для всех данных программы и фиксируется их тип (информация для этого берется из раздела описаний программы).

# Концепция данных в Паскале

- Каждому данному, используемому в программе приписывается некоторый тип.
- Тип константы определяется видом самой константы.
- Тип переменной задается явным указанием типа в разделе переменных.
- Среди типов, используемых в языке, есть стандартные и определяемые программистом.
- Любой тип определяет множество значений, которые могут принимать переменные соответствующего типа.

- Типы данных делятся на простые (скалярные) и производные.
- Значением переменной скалярного типа является одно данное (число, буква и т. п.).
- Значением переменной производного типа является, как правило, сложная структура, состоящая из различных компонентов.

- Все скалярные типы, кроме вещественных, являются порядковыми типами, т.е. среди множества их значений установлен линейный порядок.
- Для порядковых типов в Паскале определены следующие функции:
  - **Succ(x)** - дает следующее за x значение данного типа;
  - **Pred(x)** - дает предыдущее значение.

- Например: Значение **Succ(3)** равно 4, значение **Pred(7)** равно 6.
- Следует иметь в виду, что применение функции **Succ** к последнему элементу и функции **Pred** к первому элементу последовательности приводит к ошибке.

**Арифметические типы данных.**

# Данные целого типа

- В определены пять целых типов: **shortint**, **integer**, **longint**, **byte**, **word**.
- Эти типы отличаются диапазонами представимых значений, так например, в типе **integer** представимы значения от -32768 до 32767, в типе **byte** от 0 до 255.

Над целыми данными определены следующие операции:

- + сложение,
- - вычитание,
- \* умножение,
- **div** - целочисленное деление (операнды целые, результат целый),
- **mod** - остаток от деления целых чисел.

# Данные вещественного типа

Вещественные данные также представлены несколькими типами.

Наиболее употребительный из них **real**.

Над вещественными типами определены операции:

- + сложение,
- - вычитание,
- \* умножение,
- / деление.

Если операцию / применить к целым данным, то результат будет вещественный

# Стандартные функции

- Над арифметическими данными определены некоторые стандартные (элементарные) функции.
- В отличие от математики в Паскале обращение к таким функциям осуществляется единообразно, независимо от конкретной функции.
- Для использования такой функции надо написать ее имя и за ним в скобках параметр (аргумент).

Функция	Вызов функции	Тип аргумента	Тип значения
Абсолютное значение X	Abs( X )	Целый/ Вещественный	Как у аргумента
Синус X радиан	Sin( X )	Вещественный	Вещественный
Косинус X радиан	Cos( X )	Вещественный	Вещественный
Арктангенс X радиан	ArcTan( X )	Вещественный	Вещественный
Квадратный корень из X	Sqrt( X )	Целый/ Вещественный	Как у аргумента
Значение квадрата X	Sqr( X )	Целый/ Вещественный	Как у аргумента
Значение e в степени X	Exp( X )	Вещественный	Вещественный
Натуральный логарифм X	Ln( X )	Вещественный	Вещественный
Целая часть значения X	Trunc( X )	Вещественный	LongInt
Дробная часть значения X	Frac( X )	Вещественный	Вещественный
Целая часть значения X	Int( X )	Вещественный	Вещественный
Округление X	Round(X)	Вещественный	LongInt